Colophon
The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Fujitsu Semiconductor (Shanghai) Co., Ltd.
Application Note

MCU-AN- 500052-E-10

F²MC-8FX FAMILY

8-BIT MICROCONTROLLER

MB95310/370 SERIES

I2C LIBRARY USAGE API

APPLICATION NOTE

Revision History

Date	Author	Change of Records
2009-11-10	Jane Li	V1.0, First draft
2011-03-17	Jane Li	Modify Chinese remark

This manual contains 16 pages.

CONTENTS

1 Introduction

This document introduces API for I2C library.

Fujitsu MCU MB95F310 has I2C module and it can be used by connecting master device to slave device. In following chapter we will describe the library of Fujitsu MCU MB95F310.

2 I2C Library Function List

This chapter introduces all functions in I2C library in project Simulate LCD EVBoard.prj which uses MB95F310 as MCU.

Table 2-1 lists the I2C functions.

Table 2-1: I2C Functions

Function name	Description
void initial_I2C(void)	Initializes I2C and sets shift clock frequency
void Write_I2C(unsigned char addr, unsigned char subadd, unsigned char data)	Writes data to slave chip
unsigned char Read_I2C(unsigned char addr, unsigned char subaddr)	Reads data from slave chip

3 I2C Function Detail

This chapter introduces the detail of I2C functions.

3.1 Initial_I2C Function

Table 3-1 describes initial_I2C function.

Table 3-1: AD_Init Function

Function name	**initial_I2C**
Function prototype	void initial_I2C(void)
Behavior description	Initializes I2C and sets shift clock frequency
Input parameter	None
Return value	None
Example	Sets shift clock frequency to 23KHz when MCLK is 6MHz: initial_I2C();

This function is to initialize I2C. In this function default shift clock frequency is 23K when MCLK is 6MHz, and it can be decided by setting register ICCR.

The shift clock frequency can be decided by following formula:

FSCLK = MCLK / (m*n+2)

The parameter m and n are decided by register ICCR0, which is used in initial function. User can ignore it when using 6M MCLK and 23 KHz shift clock frequency.

3.2 Write_I2C Function

Table 3-4 describes Write_I2C function.

Table 3-2: Write_I2C Function

Function prototype	void Write_I2C(unsigned char addr, unsigned char subadd, unsigned char data)
Behavior description	Writes data to slave chip
Input parameter1	addr, address of slave chip, range is 0x00 to 0xff
Input parameter2	subadd, sub address of slave chip, range is 0x00 to 0xff
Input parameter3	data, data which write to slave chip, range is 0x00 to 0xff
Return value	None
Example	In case slave chip is EEPROM and it’s address is 0xa0, then write 0x25 to EEPROM sub-address 0x01: Write_I2C(0xa0,0x01,0x25);

3.3 Read_I2C Function

Table 3-5 describes Read_I2C function.

Table 3-3: Read_I2C Function

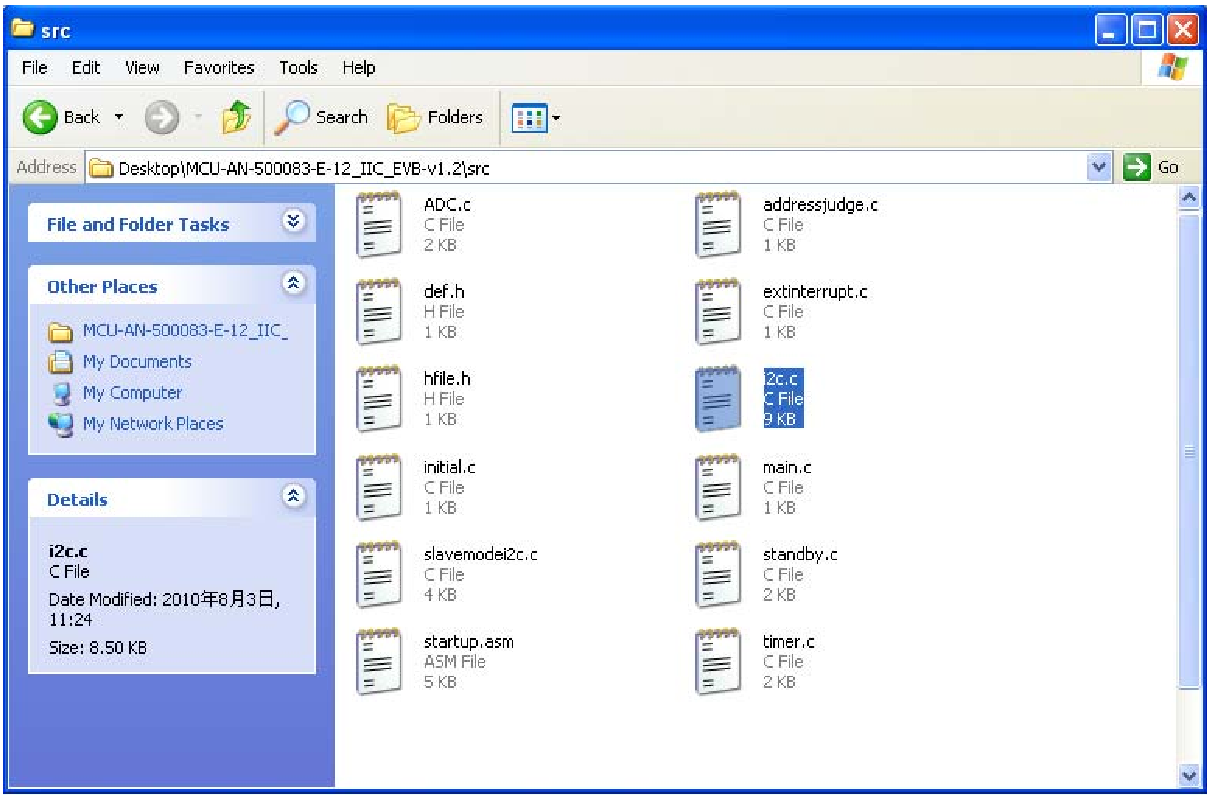
Function name	**AD_Read**
Function prototype	unsigned char Read_I2C(unsigned char addr, unsigned char subaddr)
Behavior description	Read data from slave chip
Input parameter1	addr, address of slave chip, rang is 0x00 to 0xff
Input parameter2	subadd, sub address of slave chip, rang is 0x00 to 0xff
Return value	Data saved in slave chip sub-address
Example	Read data from EEPROM and sub-address is 0x01: [variable]= Read_I2C(0xa0,0x01);

4 Usage Demo

This chapter describes the steps of how to use this library and some usage cautions.

4.1 Usage Steps of MB95F310 MCU

- ✧ First step is adding I2C library to project document, Figure 4-1 is a sample for adding this library.



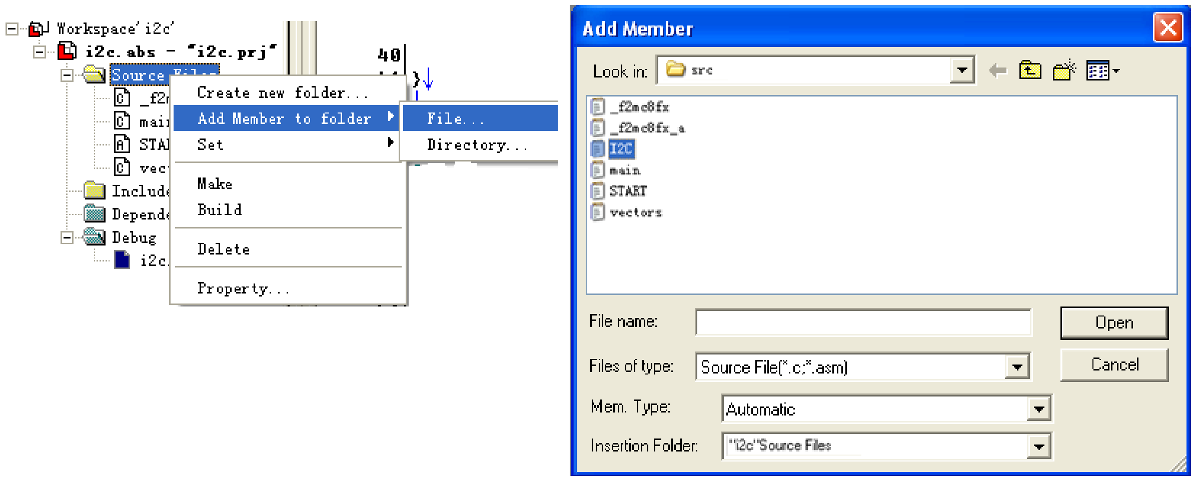
Address: Desktop\MCU-AN-500083-E-12_IIC_EVB-v1.2\src

i2c.c
C File
Date Modified: 2010年8月3日, 11:24
Size: 8.50 KB

ADC.c C File 2 KB	addressjudge.c C File 1 KB
def.h H File 1 KB	extinterrupt.c C File 1 KB
hfile.h H File 1 KB	i2c.c C File 9 KB
initial.c C File 1 KB	main.c C File 1 KB
slavemodei2c.c C File 4 KB	standby.c C File 2 KB
startup.asm ASM File 5 KB	timer.c C File 2 KB

Figure 4-1: First Step of Use I2C Library

- ✧ Second step is adding I2C library to project, Figure 4-2 is a sample for this work



File name:
Files of type: Source File(*.c;*.asm)
Mem. Type: Automatic
Insertion Folder: "i2c"Source Files

Figure 4-2: Second Step of Use I2C Library

✧ Third step is using the I2C library, for detailed library please refer to Table 2 – 1. About the initial function it is used in the library internally, so user can ignore it.

If user wants to write data, please use Write_I2C function, Figure 4 – 3 is an example which writes data 0x25 to EEPROM sub-address 0x01.

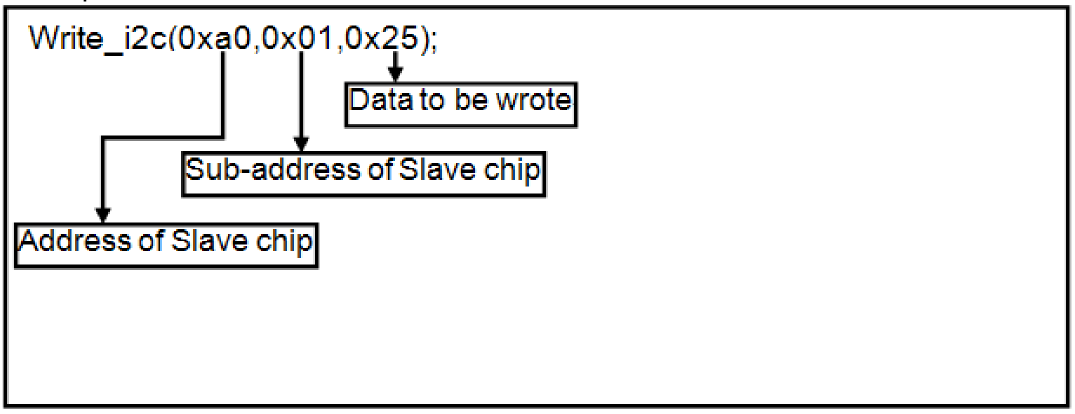


Figure 4-3: Example of Writing

If user wants to read data from slave device, Figure 4 – 4 is an example which reads data from EEPROM sub-address 0x01.

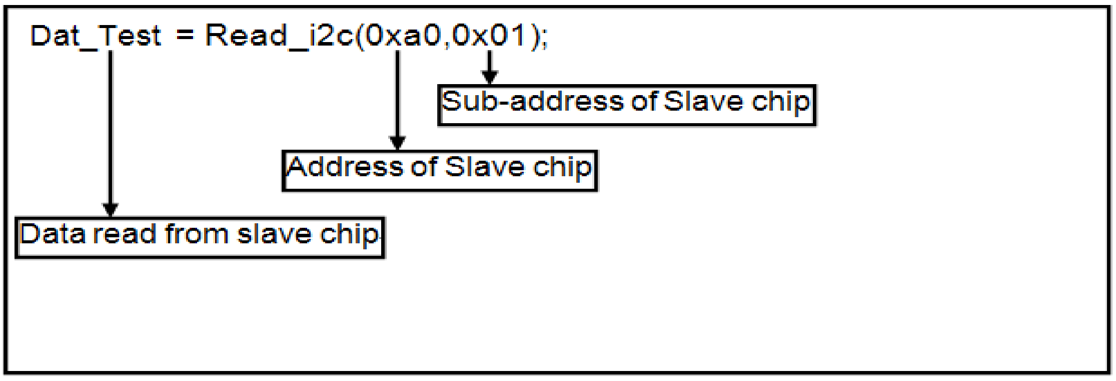
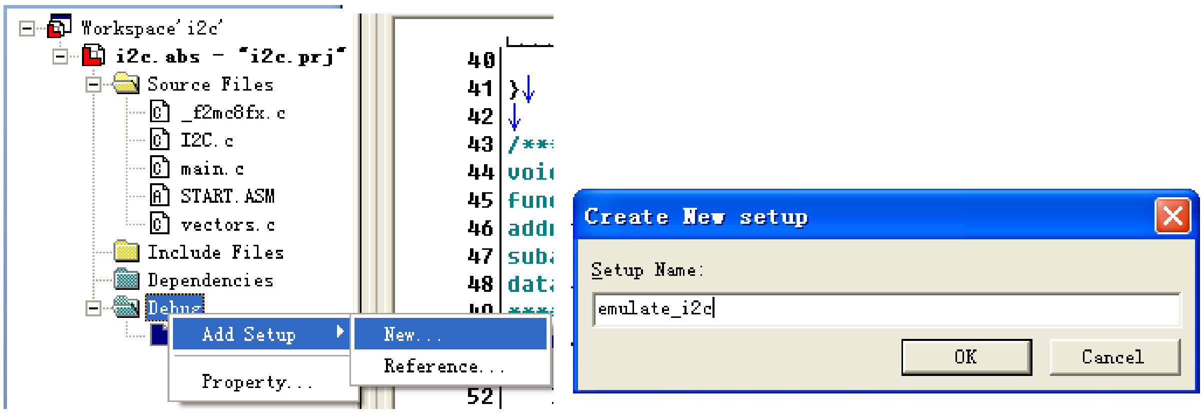

Dat_Test = Read_i2c(0xa0,0x01);
Sub-address of Slave chip
Address of Slave chip
Data read from slave chip

Figure 4-4: Example of Reading

✧ Fourth step is debugging,

User needs to add debug document. Debugging mode includes serial interface, USB interface and LAN interface. Figure 4 – 5 is an example for creating serial interface debug document.



Setup Name: emulate_i2c

Figure 4-5: Step of Setting Debug Environment

For detailed debug information please refer to Chapter 5.

4.2 Software Usage Attention

When writing data to slave device, if the slave device didn’t generate acknowledge sign to master, the following items may cause it:

- The connect of I2C is not steady
- The power of slave chip maybe not normal
- I2C is not initialized, I2C bus is busy
- Shift frequency clock is not suitable for slave chip

4.3 Sequence of Data Transfer

When transmitting, the MSB bit is transmitted first.

5 Debugging

This chapter describes how to debug the sample code I2C.prj on start kit and what will happen when the code is running.

Attachment is the sample project I2C.prj.

This project is based on our start-kit MB2146-450-E and the target MCU is MB95F310. For detailed information, you can access our website or contact with our sales window person.

When debugging, user must connect slave device and master chip by I2C. Figure 5-1 describes the connecting.

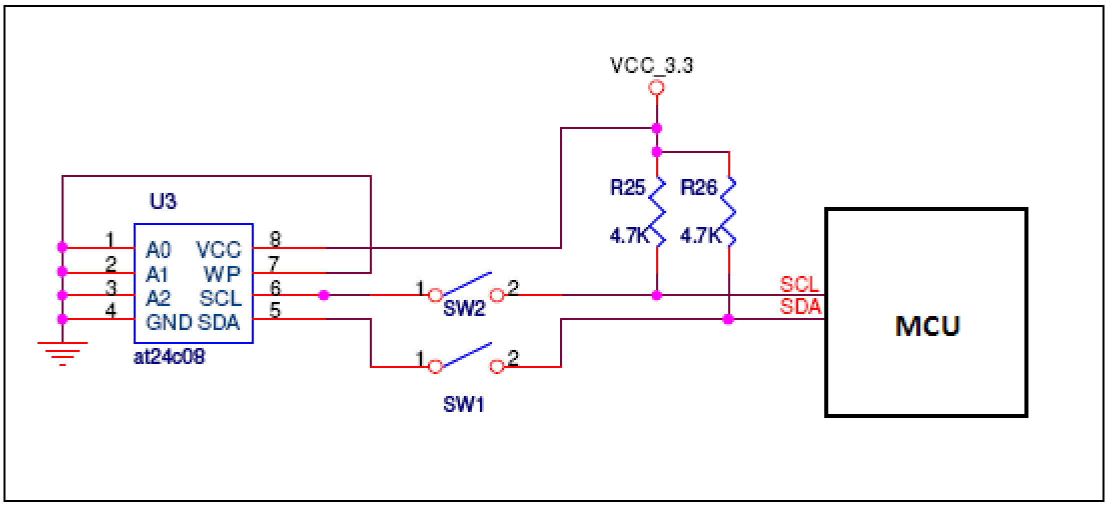
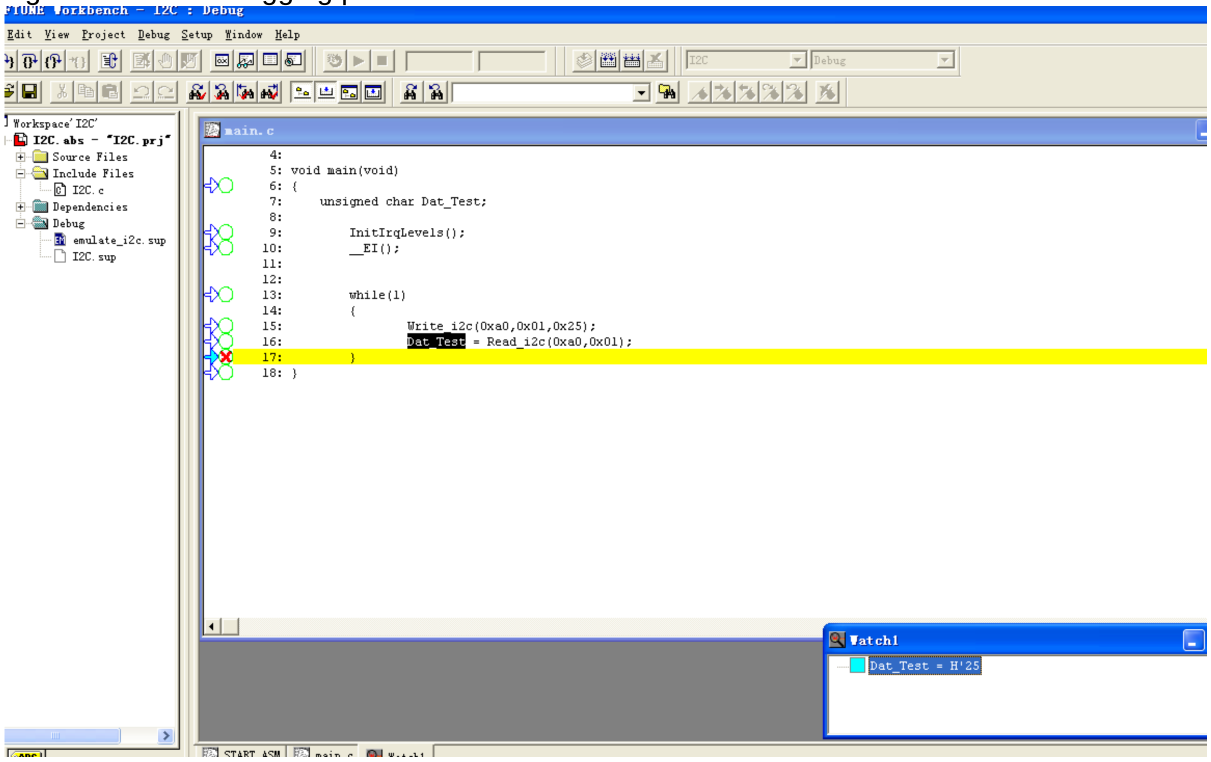


Figure 5-1: Hardware Description

The value of pull up resister R25 and R26 can select from 1KΩ to 10kΩ.

When debugging user can write or read data from EEPROM. In this project global variable Dat_Test is used to save the data read from slave device.

Figure 5-2 is a debugging picture. User can see read result in watch window.



```
void main(void)
{
    unsigned char Dat_Test;

        InitIrqLevels();
        __EI();


        while(1)
        {
                Write_i2c(0xa0,0x01,0x25);
                Dat_Test = Read_i2c(0xa0,0x01);
        }
}
```

```
Dat_Test = H'25
```

Figure 5-2: See Variable Dat_Test

6 Additional Information

For more information about how to use MB95310 EV-board, BGM Adaptor and SOFTUNE, please refer to EV-Board MB2146-450-E User Manual, or visit websites:

English version:

http://www.fujitsu.com/cn/fsp/services/mcu/mb95/application_notes.html

Simplified Chinese Version:

http://www.fujitsu.com/cn/fss/services/mcu/mb95/application_notes.html

7 Appendix